**Ruby on Rails**

Ruby on Rails, or Rails, is a server-side [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) written in [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)" \o "Ruby (programming language))under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). Rails is a [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) framework, providing default structures for a [database](https://en.wikipedia.org/wiki/Database), a [web service](https://en.wikipedia.org/wiki/Web_service), and [web pages](https://en.wikipedia.org/wiki/Web_page). It encourages and facilitates the use of [web standards](https://en.wikipedia.org/wiki/Web_standards) such as [JSON](https://en.wikipedia.org/wiki/JSON) or [XML](https://en.wikipedia.org/wiki/XML) for data transfer, and [HTML](https://en.wikipedia.org/wiki/HTML), [CSS](https://en.wikipedia.org/wiki/CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) for display and user interfacing. In addition to MVC, Rails emphasizes the use of other well-known [software engineering](https://en.wikipedia.org/wiki/Software_engineering) [patterns](https://en.wikipedia.org/wiki/Software_design_pattern) and paradigms, including [convention over configuration](https://en.wikipedia.org/wiki/Convention_over_configuration) (CoC), [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself)(DRY), and the [active record pattern](https://en.wikipedia.org/wiki/Active_record_pattern).[[4]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-4)

Ruby on Rails' emergence in the 2000s greatly influenced web app development, through innovative features such as seamless database table creations, migrations, and scaffolding of views to enable rapid application development. Ruby on Rails' influence on other web frameworks remains apparent today, with many frameworks in other languages borrowing its ideas, including [Django](https://en.wikipedia.org/wiki/Django_(web_framework)) in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Laravel](https://en.wikipedia.org/wiki/Laravel) in [PHP](https://en.wikipedia.org/wiki/PHP), Phoenix in [Elixir](https://en.wikipedia.org/wiki/Elixir_(programming_language)), and [Sails.js](https://en.wikipedia.org/wiki/Sails.js) in [Node.js](https://en.wikipedia.org/wiki/Node.js).

**ASP.NET**

ASP.NET is an [open-source](https://en.wikipedia.org/wiki/Open_source)[[2]](https://en.wikipedia.org/wiki/ASP.NET#cite_note-2) [server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) designed for [web development](https://en.wikipedia.org/wiki/Web_development) to produce [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page). It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service).

It was first released in January 2002 with version 1.0 of the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](https://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](https://en.wikipedia.org/wiki/List_of_CLI_languages). The ASP.NET [SOAP](https://en.wikipedia.org/wiki/SOAP) extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is [ASP.NET Core](https://en.wikipedia.org/wiki/ASP.NET_Core). It is a re-implementation of ASP.NET as a modular [web framework](https://en.wikipedia.org/wiki/Web_framework), together with other frameworks like [Entity Framework](https://en.wikipedia.org/wiki/Entity_Framework). The new framework uses the new open-source [.NET Compiler Platform](https://en.wikipedia.org/wiki/.NET_Compiler_Platform) (codename "Roslyn") and is [cross platform](https://en.wikipedia.org/wiki/Cross_platform). [ASP.NET MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC), ASP.NET Web API, and ASP.NET Web Pages (a platform using only [Razor](https://en.wikipedia.org/wiki/ASP.NET_Razor) pages) have merged into a unified MVC 6.[[3]](https://en.wikipedia.org/wiki/ASP.NET#cite_note-asp.net-3)

**TOP 10 2017 – OWASP**

Top 10-2017 Top 10

[**Translation Efforts - Otros Idiomas**](https://www.owasp.org/index.php/OWASP_Top_Ten_Project#Translation_Efforts_2)

|  |  |  |
| --- | --- | --- |
| [← Application Security Risks](https://www.owasp.org/index.php/Top_10-2017_Application_Security_Risks) | [2017 Table of Contents](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project)  [PDF version](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf) | [A1-Injection →](https://www.owasp.org/index.php/Top_10-2017_A1-Injection) |

|  |
| --- |
| OWASP Top 10 Application Security Risks - 2017  [**A1:2017-Injection**](https://www.owasp.org/index.php/Top_10-2017_A1-Injection)  Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.  [**A2:2017-Broken Authentication**](https://www.owasp.org/index.php/Top_10-2017_A2-Broken_Authentication)  Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.  [**A3:2017-Sensitive Data Exposure**](https://www.owasp.org/index.php/Top_10-2017_A3-Sensitive_Data_Exposure)  Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.  [**A4:2017-XML External Entities (XXE)**](https://www.owasp.org/index.php/Top_10-2017_A4-XML_External_Entities_(XXE))  Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.  [**A5:2017-Broken Access Control**](https://www.owasp.org/index.php/Top_10-2017_A5-Broken_Access_Control)  Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.  [**A6:2017-Security Misconfiguration**](https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration)  Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.  [**A7:2017-Cross-Site Scripting (XSS)**](https://www.owasp.org/index.php/Top_10-2017_A7-Cross-Site_Scripting_(XSS))  XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.  [**A8:2017-Insecure Deserialization**](https://www.owasp.org/index.php/Top_10-2017_A8-Insecure_Deserialization)  Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.  [**A9:2017-Using Components with Known Vulnerabilities**](https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)  Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.  [**A10:2017-Insufficient Logging&Monitoring**](https://www.owasp.org/index.php/Top_10-2017_A10-Insufficient_Logging%26Monitoring)  Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring. |